VeriSilicon SMIC 0.15μm Synchronous Diffusion ROM Compiler User's Guide

Trademark Acknowledgments

VeriSilicon & the VeriSilicon logo are the trademarks of VeriSilicon Microelectronics (Shanghai) Co., Ltd.

All other products and company names mentioned may be the trademarks of their respective owners.

© 2002 VeriSilicon Microelectronics (Shanghai) Co., Ltd. All rights reserved.

Printed in P.R.China.

VeriSilicon Microelectronics (Shanghai) Co., Ltd. reserves all its copy rights and other intellectual property rights, ownership, powers, benefits and rights arising or to arise from this manual. All or part of the contents of this manual may be changed by VeriSilicon Microelectronics (Shanghai) Co., Ltd. without notice at any time for any reason, including but not limited to improvement of the product relating hereto.

VeriSilicon Microelectronics (Shanghai) Co., Ltd. shall not undertake or assume any obligation, responsibility or liability arising out of or in respect of the application or use of the product described herein, except for reasonable, careful and normal uses.

Nothing, whether in whole or in part, within this manual can be reproduced, duplicated, copied, changed or disposed of in any form or by any means without prior written consent by VeriSilicon Microelectronics (Shanghai) Co., Ltd..

VeriSilicon Microelectronics (Shanghai) Co., Ltd.

3F, Building 1, No.200, Zhangheng Road, Zhangjiang Hi-Tech Park, Pudong New Area, Shanghai 201204, P. R. China

Tel : +86-21-5131-1118

Fax : +86-21-5131-1119

Web : http://www.verisilicon.com

Contents

Chapter 1 Introduction 4

1.1 Compiler Description....4
1.2 Features....4
1.3 Operating Conditions....4
1.4 Pin Descriptions....5
1.5 Parameter Range....5
1.6 ROM Floor Plan....6

Chapter 2 Timing Diagram 7

2.1 Timing Specifications for Diffusion ROM....7
Timing Parameters....7
Power Parameters....8

Chapter 3 Using the ROM Compiler 9

3.1 System Requirement....9
3.2 Software Environment....9
3.3 Installing ROM Compiler....9
3.4 Inputs and Outputs....10
3.5 Getting Started....11
Using Shell Commands....11
Using Graphical User Interface (GUI)....12
3.6 Generating the Outputs....14

chapter 1 Introduction

1.1 Compiler Description

VeriSilicon SMIC 0.15μm Synchronous Diffusion ROM compiler optimized for Semiconductor Manufacturing International Corporation (SMIC) 0.15μm Logic 1P7M Salicide 1.5/3.3V process can flexibly generate memory blocks by a friendly GUI or shell commands.

The compiler supports comprehensive range of word length and bit length. While satisfying speed and power requirements, it was optimized for area efficiency.

VeriSilicon SMIC 0.15μm LV Synchronous Diffusion ROM compiler uses three metal layers within the blocks and supports metal 4, 5, 6 or 7 as the top metal. Dummy bit cells are designed in with the intention to enhance reliability.

1.2 Features

- High Density
- High Speed
- Size Sensitive Self-time Delay for Fast Access Time and "Zero" Hold Time
- Automatic Power Down

1.3 Operating Conditions

The following table gives the recommended operating conditions for memory blocks generated by ROM compiler:

Operating Conditions	Parameter	Rating
Supply Voltage	1.5V	1.35V to 1.65V
Temperature	25ºC	0ºC to 125ºC

1.4 Pin Descriptions

The following table gives detailed information of pin descriptions for ROM (Bus index descending ordered):

Pin	Description
DOUT[i]	Data output
AD[i]	Address input
CEN	Chip enable input, low enable
CLK	CLK input, positive edge active
OE	Output enable, high enable

1.5 Parameter Range

<table>
<tr><th>Parameter</th><th colspan="3">Range</th></tr>
<tr><td>Memory Array Range</td><td colspan="3">128 to 2M Bits</td></tr>
<tr><td>Data Width</td><td colspan="3">2 to 128 Bits, Increments of 1</td></tr>
<tr><td>ROM Address Depth</td><td colspan="3">64 to 32768 Words, Increments of (8 X Column Mux)</td></tr>
<tr><td rowspan="5">Column Mux</td><td colspan="3">The following lists the changes of width and height when column mux is set to different values. Suppose when column mux is 16, the width and height is a standard.</td></tr>
<tr><td></td><td>Width</td><td>Height</td></tr>
<tr><td>8</td><td>1/2</td><td>2</td></tr>
<tr><td>16</td><td>1</td><td>1</td></tr>
<tr><td>32</td><td>2</td><td>1/2</td></tr>
<tr><td>Top Metal</td><td colspan="3">m4, m5, m6 or m7</td></tr>
<tr><td>Output Drive Strength</td><td colspan="3">The same drive as INVHD8X cell in VeriSilicon SMIC 0.15μm High-Density Standard Cell Library</td></tr>
</table>

1.6 ROM Floor Plan

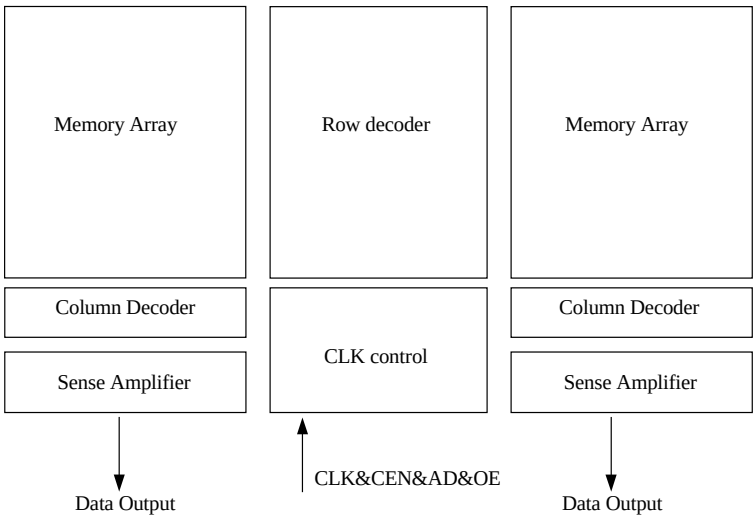


Fig.1 ROM Floor Plan

chapter 2 Timing Diagram

2.1 Timing Specifications for Diffusion ROM

This section specifies the timing specifications for the diffusion ROM.

Fig.2 gives the read function timing diagrams of diffusion ROM.

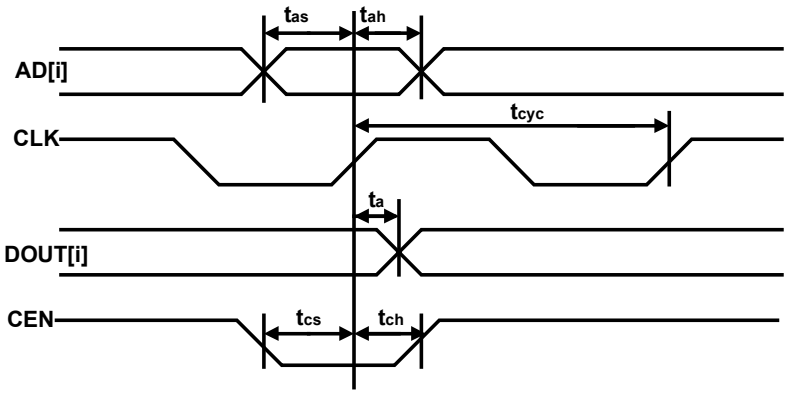


Fig.2 Read-Function Timing

Fig.3 gives the diffusion ROM output enable timing diagrams:

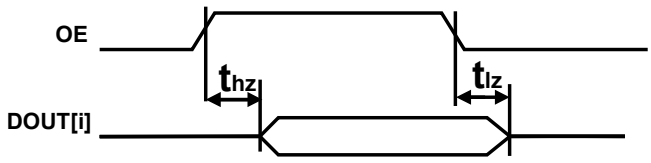


Fig.3 Output-Enable Function Timing

Timing Parameters

The following table specifies the timing parameters in the datasheet generated by the ROM compiler:

Timing Parameters

Parameter	Symbol
Cycle time	tcyc
Access time*	ta

Address setup time	tas
Address hold time	tah
Chip enable setup time	tcs
Chip enable hold time	tch
Output enable to hi-Z time	t_{lz}
Output enable active time*	t_{hz}

* Parameters values are load dependent.

Power Parameters

The following table specifies the power parameters in the datasheet generated by the ROM compiler:

Power Parameters

Parameter	Symbol
Average current	Iavg
Peak current	Ipeak

i_{avg} is the average current in A/100MHz unit. The average current in the datasheet is achieved under below assumption:

1. Input net transition is 0.2ns.
2. Output port capacitance is 0pF.

The total average current of the memory than can be estimated according following equation:

$I_{avg} = i_{avg} * F + 1/2 * C * V * f * N$

Where,

I_{avg}: the total average current of the memory. (A)

F: the frequency of clock. (100MHz)

C: the average capacitance of output port. (F)

V: the voltage supply. (V)

f: the frequency of output port. (Hz)

N: switched ports number.

i_{peak} is the peak current of memory during operation in A unit.

chapter 3 Using the ROM Compiler

3.1 System Requirement

Before installation, make sure that the following minimum host configuration is available:

- Sun Microsystem's Solaris7

3.2 Software Environment

The ROM compiler requires UNIX and X-Window as its GUI was developed with Motif.

3.3 Installing ROM Compiler

To install ROM compiler, please follow the instructions bellow:

1. Create an installation directory where you wish to install the ROM compiler.

> ***NOTE***
> *<install_dir> will stand for the directory you have created for installation hereafter.*

2. cd <install_dir>
3. gunzip < <release_compressed_file> | tar xvf -
4. Copy .vsmcrc file to the home directory.
5. Add the following to .cshrc file.

   source ~/.vsmcrc
6. Modify .vsmcrc file as the following and source it:

   setenv VERISILICON_MC_DIR <install_dir>

After successful installation, the following directory structure will be created under <*install_dir*>:

drom.15

This directory contains the technology files and library files of the ROM

compiler.

The following table lists the names of the executable files in the installation directory and its description:

Name	Description
MC	The executable file of ROM compiler.

NOTE

Be sure not to edit any files in <install_dir> *directory*.

3.4 Inputs and Outputs

The ROM compiler allows users to define the following input parameters for a specific ROM block:

Library

Running Directory

Block Name

Number of Words

Number of Bits

Ring Width

Frequency (MHz) (no use now)

ROM Code File

Multiplexer Width

Horizontal Ring Layer

Vertical Ring Layer

Top Metal Layer

The ROM compiler generates the following output files.

GDSII Layout File (GDSII format)

LVS Netlist (CDL format)

Verilog Model Code

VHDL Model Code

TLF Timing

Synopsys Model

Datasheet

ROM code file

LEF view

Antenna LEF view

Antenna CLF model

After inputting the parameters, users should wait a few minutes for the outputs to be generated by the ROM compiler automatically.

3.5 Getting Started

There are two ways to start ROM compiler as follows:

Using Shell Commands

Users can launch ROM compiler using commands line in the shell window. Enter the following commands to launch the ROM compiler directly from the shell:

% cd <running_dir>

% MC [options with parameters]

The <*running_dir*> is the directory which the ROM compiler run in. All the outputs will be generated in this directory.

Make sure that the running directory < *running_dir*> is different from the installation directory <*install_dir*>.

The following options can be specified in the command line:

-lib $lib_dir

-outdir $run_dir

-block $mem_name

-wordsnumber $memlength

-bitsnumber $datawidth

-ringwidth $ringwidth

-muxwidth $varMuxWidth

-vlayer $varVLayer

-hlayer $varHLayer

-frequency $frequency (no use now)

-codefile $codefile

-topmetal $topmetal

-area y/n

Please see section **Parameters under GUI and Shell Commands** for details explanation.

Example:

MC -lib drom.15 -block DROM2048X16 -wordsnumber 2048 -bitsnumber 16 -muxwidth 16 -vlayer m3 -hlayer m4 -frequency 100 -codefile DROM2048X16 -topmetal m4

This command will generate a ROM name of DROM2048X16, with 2048 words, 16 bits, 16 column multiplexer width, frequency 100MHz, metal layer 3 as the vertical ring layer, metal layer 4 as the horizontal ring layer, and metal layer 4 as the top metal layer according to the code file DROM2048X16.

Using Graphical User Interface (GUI)

We provide a friendly GUI to enable the users to configure parameters and generate all the outputs in the directory specified. From the shell, type the commands as follows:

% cd <*running*_dir>

% MC

Click on the *browse* button to select the ROM's library, then the following GUI window for ROM compiler will appear on your screen:

Fig.3 the ROM Compiler GUI

Fill content in the blank for each option and click the proper button, you will get your results.

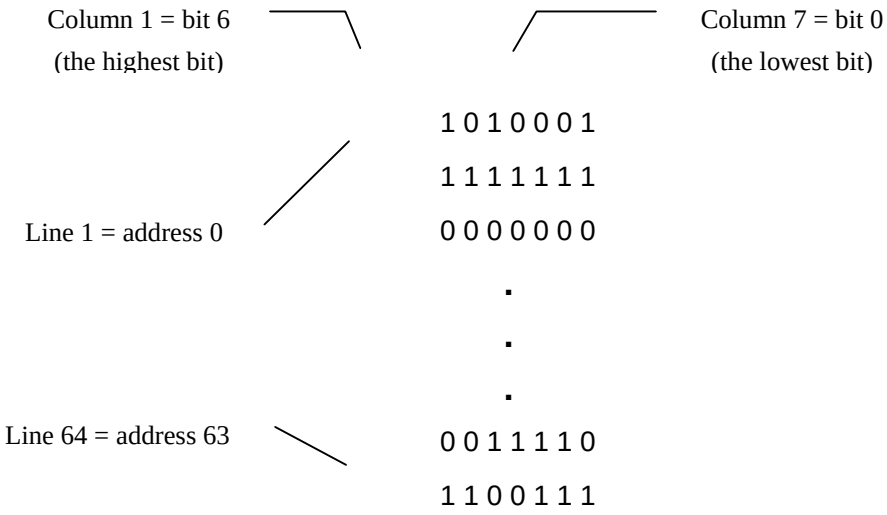
Parameters under GUI and Shell Commands

The section specifies detailed descriptions of the parameters of the ROM compiler and their corresponding default value:

<table>
<tr><th>Parameter under Shell Commands</th><th>Parameter under GUI</th><th>Description</th></tr>
<tr><td>-lib</td><td>Library</td><td>Specify the library directory used by the Register File compiler.
You can click on the browse button to find the valid library you have installed.</td></tr>
<tr><td>-rundir</td><td>Running Directory</td><td>Specify the output directory of the ROM compiler.
The directory name can be any valid path name supported by the system. The default is current directory.</td></tr>
<tr><td>-block</td><td>Block Name</td><td>Specify the block name.
The block name can include any alphanumeric value and must be unique to avoid name conflicts for blocks within the same library. It is recommended that a block name is no more than 16 characters, for we will identify two blocks by their first 16 letters.
The default is DROM4096X32M8.</td></tr>
<tr><td>-wordsnumber</td><td>Number of Words</td><td>Specify the number of words in the block.
The default value is 4096. The range for words can be 64 to 32768.
The following gives the detailed information:
<table>
<tr><th>Mux</th><th>Number of words</th><th>Increment</th></tr>
<tr><td>8</td><td>64 to 8192</td><td>Mux * 8</td></tr>
<tr><td>16</td><td>128 to 16384</td><td>Mux * 8</td></tr>
<tr><td>32</td><td>256 to 32768</td><td>Mux * 8</td></tr>
</table></td></tr>
<tr><td>-bitsnumber</td><td>Number of Bits</td><td>Specify the number of bits in the block.
The default value is 32. The range for bits can be 2 to 128.
The following gives the detailed information:
<table>
<tr><th>Mux</th><th>Number of Bits</th><th>Increment</th></tr>
<tr><td>8</td><td>2 to 128</td><td>1</td></tr>
<tr><td>16</td><td>2 to 128</td><td>1</td></tr>
<tr><td>32</td><td>2 to 64</td><td>1</td></tr>
</table></td></tr>
<tr><td>-ringwidth</td><td>Ring Width</td><td>Specify the ring width of the block in um.
The default value is 5. The minimum is 2. The designer must decide the ring width according the power analysis.</td></tr>
<tr><td>-frequency</td><td>Frequency</td><td>Specify the frequency of the clock of the chip in MHz. The default value is 100MHz. It dose not affect anything now, and it is just reserved for future usage.</td></tr>
</table>

-codefile	**CodeFile**	Specify the code file of the ROM. Please refer to the following figure for detailed information about the code file format. If customer doesn't specify the code file or the code file he specifies doesn't exist, all the bits of the ROM block will be set to 1 automatically. The default is NOCODE.
-muxwidth	**Multiplexer Width**	Specify the column multiplexer width. The default value is 8. There are three buttons for your choice: 8, 16, or 32. When this option is set to different values, the width and height of the block will change correspondingly. For detailed information, please refer to Parameter Range section on page 5.
-vlayer	**Vertical Ring Layer**	Specify which metal layer will be the vertical ring layer. It can only be m4.
-hlayer	**Horizontal Ring Layer**	Specify which metal layer will be the horizontal ring layer. It can only be m3.
-topmetal	**Top Metal Layer**	Specify the top metal layer. It can be m4, m5 or m6. The default value is m4.
-area	**y/n**	Once "–area y" is used, the compiler will only generate a width X height report. Default is n.

The following illustrates the format of code file of a ROM name of DROM64X7, with 64 words, 7 bits:


Column 1 = bit 6
(the highest bit)
Column 7 = bit 0
(the lowest bit)
1 0 1 0 0 0 1
1 1 1 1 1 1 1
Line 1 = address 0
0 0 0 0 0 0 0
Line 64 = address 63
0 0 1 1 1 1 0
1 1 0 0 1 1 1

Each character of a line indicates the bit of a word.

3.6 Generating the Outputs

When you click on the *Default* button in the GUI window, the ROM compiler will automatically load the default parameters of the ROM and generate the ROM based on the default parameters.

To generate the outputs, click on *Generate* button. All the outputs are

generated according to the generic parameters you set and place in the user specify *running* directory <*running_dir*>.

The following table lists the detailed description of the output files:

Name	Description
*.tlf	TLF Model
*.lib	Synopsys Model
*.net	Cdl netlist
*.gds	GDS file
*.ds	Datasheet
*.v	Verilog Model
*.vhd	VHDL Model
*.dat	Rom Code File
*.lef	LEF view
*_antenna.lef	Antenna LEF view
*_antenna.clf	Antenna CLF model

And click on *Exit* button to quit the ROM compiler.